

Input Specification

```
data double x(l);
x~gauss(mu,sigma);
...
```

Synthesis System

C++

```
for( i= 0;
i < n_points;
i++ ){
...
}
```

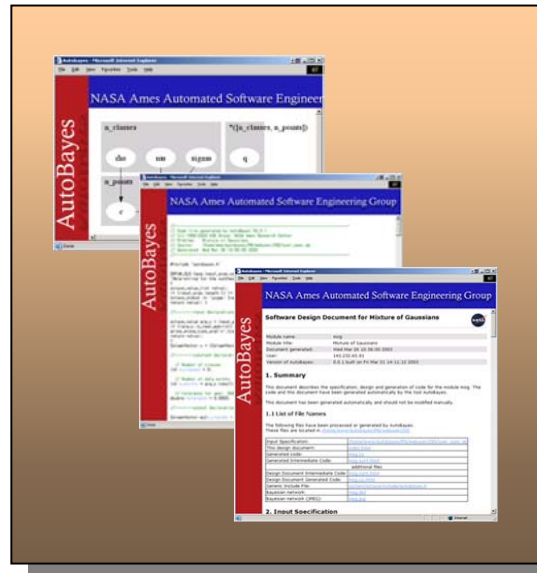
Problem: *Generation of Code alone is not enough*

Automatic program synthesis systems can generate complex C/C++ code to solve NASA-relevant problems. This code must be documented and understandable for review, integration, and QA purposes.

Solution: *Automatically generate:*

- Documented code
- Standardized Software Design Document

Software Design Document



- Hyperlinked document in accordance with important NASA and IEEE standards
- Browsable specification and code
- Concise interface description
- Documentation of assumptions and design information

Explanation of Accomplishment

- **POC:** Phil Oh, (QSS, ASE, Code IC, oh@email.arc.nasa.gov), Johann Schumann, (RIACS, ASE, Code IC, schumann@email.arc.nasa.gov), Bernd Fischer (RIACS, ASE, Code IC, fisch@email.arc.nasa.gov)
- **Background:** Autocoding techniques promise large gains in software development productivity. Program synthesis systems can generate large C++ programs that solve NASA-relevant problems, such as those used in guidance applications. However, the real-world application of autocoding technology has been limited, particularly in areas where a rigorous software development process is used. Because the automatically generated code needs to be tested, integrated, and certified, synthesis of code alone is not sufficient: the generated code must be readable and understandable.
- **Accomplishment:** For practical usability, we have extended our generator of Software Design Documents (SDD). The previous version worked with the deductive synthesis system Amphion/NAV, the new version works with the schema-guided synthesis systems AutoFilter/AutoBayes. The hyperlinked design document contains detailed information about the specification, the interface definition for the synthesized code, and now explicitly: the assumptions in the generated program (e.g., number of data points is larger than 0); the limitations of the program; the error messages the program can produce; and any outstanding proof obligations. When relevant, a graphical representation of the input specification is produced. The intermediate code and the final generated code are displayed in a color-coded scheme and allows the user to navigate and trace between specification and generated code. A sample SDD is at <http://ase.arc.nasa.gov/autobayes/SDD>
- **Benefits:** This technology has the potential to substantially increase the use of autocoding technology within NASA, as the generated standardized and detailed design document facilitates integration of the synthesized code into a rigorous software development process including code review, testing, and certification.
- **Future Work:** As part of product-oriented certification, the SDD will be extended to contain certification results.